BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

CLAIMS

WHAT IS CLAIMED IS:

1. A memory system for converting virtual addresses into physical addresses at the user code level and conveying the physical addresses to a peripheral device, the system comprising:

a virtual memory having memory pages, each memory page having a virtual memory address and a corresponding physical memory address in a main memory;

a microprocessor having a translation look-aside buffer which holds a subset of the virtual addresses for all the pages and their corresponding physical memory addresses in the main memory; and

a set of user-code level software instructions stored in said main memory and executable by said microprocessor for converting a virtual memory address into the corresponding physical memory address, pinning the page, and conveying the physical memory address to the peripheral device at a user code level without requiring a system call.

2. The memory system of claim 1, wherein the set of user-code level software instructions directs the microprocessor to utilize the translation look-aside buffer (TLB) in order to convert the virtual memory address into its corresponding physical memory address and pin the page.

3. The memory system of claim 1, wherein the set of user-level software instructions include instructions for unpinning the page, which

corresponds with the physical memory address conveyed to the peripheral device, when the peripheral device has completed its operations.

4. The memory system of claim 1, further comprising a page pin counter, which is incremented each additional time a page is pinned down after a first time said page is pinned down.

5. The memory system of claim 1, wherein the set of user-level software instructions include instructions for unpinning the page in memory, by:

decrementing the page pin counter each time the page is to be unpinned; and

unpinning the page which corresponds with the physical memory address conveyed to the peripheral drive, once the page pin counter has reached zero.

6. The memory system of claim 1, wherein the set of user level software instructions include instructions for:

incrementing a per-process pin page counter each time a new page is first pinned by the same process; and

incrementing a system wide pin page counter, each time a new page is first pinned by any process within the system.

7. The memory system of claim 6 wherein the set of user level instructions including instructions for:

checking the values in the per-process pin page counter and the system wide pin page counter, each time a new page is to be pinned within the system,

not pinning the new page associated with the physical address, if either of these counters contains a value which exceeds a maximum allowable value, otherwise pinning the new page.

8. A method for converting a virtual address to a physical address and conveying the physical addresses to a peripheral device at the user code level, the method comprising:

executing user code and encountering an instruction which requires the peripheral device to perform a memory operation, the instruction containing a command which describes the necessary operation, a virtual address of the peripheral device which is to perform the memory operation, and a virtual memory address which identifies a page where the operation is to occur; and

executing a special user-code level instruction comprising the further steps of:

translating the virtual memory address, where the operation is to occur, into a physical memory address location using a user code level instruction set; and

transferring the command, which describes the memory operation, and the physical memory address location to the peripheral device via a bus transaction, wherein the data portion of the bus transaction will contain the physical memory address location.

9. The method of claim 8, wherein the step of translating includes:

converting the virtual address which identifies a page where the operation is to occur into a corresponding physical memory address location using a translation look-aside buffer resident in the microprocessor; and

pinning down the page which corresponds with the physical memory address location.

10. The method of claim 9 wherein the translation look-aside buffer has a number of entries, each entry containing a virtual address and a corresponding physical address for a page.

11. The method of claim 9 wherein the step of pinning down the page includes setting an extra bit space in a relevant entry in the translation look-aside buffer which contains the physical memory address location which is to be transferred to the peripheral device.

12. The method of claim 11, wherein a page pin counter is incremented if the extra bit space in the relevant entry in the translation look-aside buffer which cotains the physical memory address location transferred to the peripheral device has previously been set.

13. A method of claim 11 wherein the main memory contains a page table which identifies all the pages in the main memory and their corresponding virtual and physical memory addresses.

14. The method of claim 13, wherein the setting of the extra bit space in the translation look-aside buffer, in order to pin the page, is

automatically written back into the page table so the page table also indicates that the page has been pinned.

15. The method of claim 10 wherein the step of executing the special user-code level instruction further includes:

unpinning the page, which corresponds with the physical memory address conveyed to the peripheral drive, once the peripheral device has completed its operations.

16. The method of claim 15, wherein the step of unpinning the page includes:

decrementing a page pin counter which indicates a total number of times the page to be unpinned has been previously pinned;

clearing an extra bit space in a relevant entry in the translation look-aside buffer which contains the virtual address of the page to be unpinned, once the page pin counter has reached zero.

17. The method of claim 8 wherein the step of executing the special user-code level instruction further includes:

defining a physical address port space which can only be written to by user-level code when using the special instruction, the physical address port space holding the physical address of the peripheral device that is to perform the memory operation.

18. A set of user-code level software instructions executable by a microprocessor for converting a virtual address to a physical address, pinning an associated page, and conveying the physical address to a peripheral device without requiring a system call, the instructions comprising:

a Trans_and_pin instruction for converting the virtual address to a physical address, pinning the page associated with the physical address and conveying the physical address to the peripheral, the Trans_and_pin instruction including:

a <vaddr> source operand containing the virtual address to be converted; and

a <target> source operand containing the address of the peripheral device to which the physical address is to be conveyed.

19. The set of user code level instructions of claim 18 wherein the Trans_and_pin instruction increments a page pinned counter each time the page associated with the physical address is pinned.

20. The set of user code level instructions of claim 18 wherein the Trans_and_pin instruction:

increments a per-process pin page counter each time a new page is first pinned by a process; and

increments a system wide pin page counter, each time a new page is first pinned within the system.

21. The set of user code level instructions of claim 20 wherein the Trans_and_pin instruction checks the per-process pin page counter and the system wide pin page counter, each time a new page is first pinned within the system, the Trans_and_Pin function not pinning the page associated with the physical address, if either of these counters contains a value which exceeds a maximum allowable value.

22. The set of user-code level instructions of claim 18 wherein the Trans_and_pin instruction further includes:

an <outcome> destination operand containing information which indicates whether the conversion of the virtual address has been executed successfully or the reasons for failure, if any.

23. The set of user-code level instructions of claim 18, further comprising:

a Trans_and_pin_only instruction for converting the virtual address to a physical address, pinning the page associated with the physical address and storing the physical address in a register location, the Trans_and_pin_only instruction including:

a <vaddr> source operand containing the virtual address to be converted; and

a <phy_addr_reg> destination operand containing a register location where the physical address is to be stored.

24. The set of user-code level instructions of claim 23 wherein the register location specified in <phy_addr_reg> can only be written to in user-level code by using the Trans_and_pin_only instruction.

25. The set of user-code level instructions of claim 24, wherein the register location contains a valid bit which is set after the Trans_and_pin_only instruction stores the physical address in the register location.

26. The set of user code level instructions of claim 23 wherein the Trans_and_pin_only instruction increments a page pinned counter each time the page associated with the physical address is pinned.

27. The set of user-code level instructions of claim 23 wherein the Trans_and_pin_only instruction further includes:

an <outcome> destination operand containing information which indicates whether the conversion of the virtual address to the physical address has been executed successfully or the reasons for failure, if any.

28. The set of user-code level instructions of claim 23 and 25 further comprising:

a send_physical instruction for transferring the contents of the register location specified in the Trans_and_pin_only instruction to a peripheral device, the send_physical instruction including:

a <phys_addr_reg> source operand which specifies the register location; and

a <target> source operand containing the address of the peripheral device.

29. The set of user code level instructions of claim 28 wherein the send_physical instruction checks the valid bit in the register location specified in <phys_addr_reg> to make sure it is set before transferring the contents of the register location to the peripheral device.

30. The set of user code level instructions of claim 18 and 23, further comprising:

an unpin instruction for unpinning the page previously pinned by the Trans_and_pin or Trans_and_pin_only instruction once the peripheral device is done using the page, the unpin instruction including:

a <vaddr> source operand containing the virtual address of the page to be unpinned.

31. The set of user code level instructions of claim 19 and 26 wherein the unpin instruction decrements the page pinned counter each time the page is to be unpinned and then actually unpins the page only when the contents of the page pin counter is zero.